NRL Report 7924

Fast Convolution by Number Theoretic Transforms

LAWRENCE M. LEIBOWITZ

Digital Applications Branch
Special Projects Organization
Office of the Director of Research

September 12, 1975

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 7924	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FAST CONVOLUTION BY NUMBER THEORETIC TRANSFORMS		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL Problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lawrence M. Leibowitz		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem K08-03
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		12. REPORT DATE September 12, 1975
		13. NUMBER OF PAGES 21
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Chirp z-transform (CZT)
Circular convolution
Complex integers
Complex number theoretic transform (CNT)
Discrete convolution
Discrete Fourier transform (DFT)
Fast convolution
Fast Fourier transform (FFT)
Fermat number transform (FNT)
Finite field

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The concepts involved in the efficient computation of discrete convolution by number theoretic techniques were reviewed using key portions of the applicable technical literature. First, discrete convolution and its computation by direct and fast Fourier transform (FFT) techniques are reviewed. The discrete Fourier transform (DFT), defined in a finite field or ring, and applicable number theory concepts, are used to describe the number theoretic transform (NTT) and its circular convolution properties. The considerations in the selection of a modulus that results in efficient modular arithmetic on a binary computing device are discussed. The limitations in transform length

Continued

19. Continued

Finite ring
Mersenne number transform (MNT)
Modular arithmetic
Number theoretic transform (NTT)
Number theory

20. Continued

are presented, along with means for overcoming this restriction. The complex number theoretic transform (CNT) defined in both a finite field and ring is described. Finally, the computational efficiency of NTT vs FFT convolution is discussed.

CONTENTS

FAST CONVOLUTION BY NUMBER THEORETIC TRANSFORMS

INTRODUCTION

Many signal processing applications previously performed by continuous systems can now be satisfied by means of discrete systems described by digital signal processing theory. A description of this theory appears in Ref. 1, which is considered necessary background to the material discussed here. As a result of investigation into digital signal processing theory, several factors become apparent. Although discrete convolution is a basic means of discrete system representation, its application to real-time processing is limited because of the number of arithmetic operations involved in its computation. The arithmetic round-off error inherent in the practical implementation of discrete systems generates noise in the output of digital signal processing systems. The application of number theoretic concepts utilizing modular arithmetic to the computation of discrete convolution under certain conditions provides some degree of solution to these limitations. It is these concepts in the form of number theoretic transforms (NTTs) which are discussed in this report.

DISCRETE CONVOLUTION

The output of any linear time-invariant discrete system can be expressed most generally in the form of the convolution sum or discrete convolution

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k).$$

Here $x(n)$ is the input sequence, $h(n)$ is the unit-sample (impulse) response of the system and $y(n)$ is the output sequence. In practical applications discrete convolution must be expressed in a finite form

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) = h(n) * x(n)$$

where the unit-sample response is finite or can be suitably approximated as such. Evaluation of this sum requires a total of N^2 multiplications, which gives some measure of computational complexity. Discrete convolution can be applied to systems of infinite unit-sample response, but sectioning [2,3] or block recursion techniques [4,5] must be

Note: This report represents a part of the research performed under the Edison Memorial Fellowship in partial fulfillment of the requirements for the degree of Doctor of Science at the George Washington University School of Engineering and Applied Science.

Manuscript submitted July 29, 1975.

used. In sectioning, the system output is expressed as a sum or sequence of finite convolutions. With block recursion, the input, output, and unit-sample response are partitioned into blocks of finite length and the system is represented by convolution in matrix form. The discrete convolution form of discrete system representation has been generally limited to the realization of finite unit-sample response, or FIR, filters.

A discrete convolution may be efficiently computed by application of the discrete form of the convolution theorem [6], whereby convolution in the time domain corresponds to multiplication in the frequency domain, using methods proposed by Stockham [2]. The discrete Fourier transform (DFT) [7],

$$x(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad W_N = e^{-j(2\pi/N)}, \qquad 0 \leqslant k \leqslant N-1,$$

is applied in the form of the fast Fourier transform (FFT) [8,9] to translate the input and unit-sample response sequences to the frequency domain as $X(k)$ and $H(k)$, respectively. The FFT computation of each DFT requires on the order of $N \log_2 N$ complex multiplications as opposed to N^2 for direct evaluation. The FFT algorithms most generally require that N be an integral power of 2, but other forms for highly composite N are available. The discrete convolution is obtained as the time domain representation, or inverse discrete Fourier transform (IDFT),

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) W_N^{-kn}, \qquad 0 \leqslant n \leqslant N-1,$$

applied in the form of the FFT, of the frequency-domain product $Y(k) = X(k)H(k)$. The resulting convolution is termed a circular convolution, as the DFT is applicable only to periodic functions; thus, the sequences to be convolved must be represented as periodically extended with discrete sample indices evaluated modulo N. (Modulo evaluation is defined in the section on Arithmetic Modulo M.)

Circular convolution can be used to perform linear convolution of a periodic sequence by appending sufficient zero-valued samples to each of the sequences to be convolved as necessary to avoid periodic interference, or overlap, error. To convolve a sequence of length P with a sequence of length Q, using a power of 2 FFT, requires that enough zero-valued samples be appended such that $N = 2^m \geqslant P + Q - 1$. Since, in general, the convolution computation requires two transforms, a multiplication of sequences and an inverse transform, the number of complex multiplications is on the order of $3N \log_2 N + N$ as compared to N^2 for direct evaluation. For $N \geqslant 32$, FFT convolution is much more efficient than direct evaluation.

The discrete form of correlation is

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n+k),$$

and because of the similarity in form, the above discussion for discrete convolution applies with minor differences [7]. Thus, efficient computational schemes with respect to discrete convolution are important. They are applicable to realization of both FIR and infinite impulse response (IIR) digital filters [3] as well as the statistical analysis of signals including autocorrelation and spectral estimation [10].

Because of the finite representation of all quantities represented in a digital electronic computing device, all discrete convolution and correlation computations will generally possess some error due to roundoff. There are also additional errors caused by the finite quantization of the signal and unit-sample response sequences. This quantization error in the input sequences is independent of the computation algorithm used. In the FFT realization of discrete convolution, the resulting arithmetic roundoff error or noise in the output is primarily dependent on that generated in each of the three applications of the FFT. The output noise due to arithmetic roundoff in the FFT has been analyzed both for fixed- [11] as well as floating-point [12-14] number representations. In the fixed-point case, with proper scaling to prevent overflow, the upper bound on noise-to-signal ratio is found to increase as $\sqrt{N}$, or 1/2 bit per stage, where the number of stages is $m = \log_2 N$. The floating-point implementation round-off error analyses indicate noise-to-signal ratios increasing linearly with m for fixed mantissa length.

From the above discussion it can be seen that discrete convolution and correlation are powerful tools in digital signal processing applications and can be readily and efficiently implemented by FFT techniques. However, the amount of computation required even with FFT convolution is still significant, and the use of these techniques is limited in many applications such as communications and radar where real-time processing is a requirement. Another limitation of FFT convolution is the output noise caused by required roundoff in the arithmetic processes. This noise can become quite significant for large values of N.

In recent years, several transform techniques have been proposed to more efficiently compute discrete convolutions. One such technique [15] involves the Walsh transform and requires $2 \cdot 3^{m-1}$ multiplications. This technique still involves round-off error and is slower than FFT convolution for $N > 2^9$. The main drawback of this technique is that the results of all intermediate computations must be maintained in memory in order to make the final computation. For $N = 2^9$ this would require 13,122 storage locations for each sequence to be convolved. More recently, another class of transform techniques, which involve number theory and under certain conditions overcome many of the limitations of FFT convolution, has been proposed [16-20] and will be considered here.

THE DFT IN A FINITE FIELD OR RING

The DFT, implemented in the form of the FFT, is important in the efficient computation of discrete convolution and correlation. If $\alpha = W_N = \exp(-j2\pi/N)$, the form of a DFT of a sequence $x(n)$, $n = 0, 1, ..., N - 1$ can be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n)\alpha^{nk}, \qquad k = 0, 1, ..., N-1.$$

If values of α other than the complex roots of unity are considered, other transforms may exist which can be said to be of this DFT form. Of these transforms it is those which possess the circular convolution property (CCP) (i.e., that the product of the transforms of two sequences is the transform of their circular convolution), which are of interest here. It will be shown here that the properties generally required of such transforms, when defined in a field, are that α be of order N that is N is the least positive integer such that $\alpha^N = 1$, and that N^{-1} be defined [18].

The DFT with $\alpha = \exp(-j2\pi/N)$ is the only transform defined in the field of complex numbers that possesses the CCP [18]. Pollard [16] defined the DFT in a finite field as well as in the ring of integers modulo an integer M where there are constraints on α in addition to that mentioned above. Such a ring, denoted Z_M, consists of the set of integers $\{0, 1, ..., M-1\}$ with addition, subtraction, and multiplication modulo M. A ring of integers modulo M is a field only if multiplicative inverses exist for all nonzero elements and thus division is defined, since $b/a = ba^{-1}$. This is the case only for M a prime. A field then is a special case of a ring and can, in general, be referred to as a ring of integers modulo M.

The transforms considered here are termed number theoretic transforms (NTT). In order for such transforms and their inverses to exist, there is a set of constraints that must be satisfied among the roots of unity α, convolution length N, and modulus M. There are many different NTTs that can be defined in various fields and rings. For certain values of α, N, and M it is possible to compute transforms in a highly efficient manner with each multiplication replaced by a single binary word shift and an addition. Thus, in effect, no multiplications are required. Also, if N is highly composite, an FFT type of algorithm can be applied to further reduce the amount of computation required. It is those NTT for which possibilities of efficient computation, relative to FFT convolution, exist that are of primary interest here. Another important reason for interest in these transforms is that round-off or truncation and associated errors, inherent in normal arithmetic, have no meaning in modulo arithmetic. All data and results are exactly represented among a finite set of quantities. Thus there is no noise from arithmetic roundoff in the outputs of convolutions implemented by means of NTT, provided certain conditions, to be given later, are met. In addition, unlike the powers of W_N used in the FFT, for certain values of α there is no need to store the basis functions (powers of α) of the NTT.

Practical applications of discrete convolution, in digital signal processing, involve amplitude data that do not generally belong to a finite field or ring. How then does the existence of transforms with the CCP in finite fields or rings aid in the computation of convolution in digital filtering and other practical areas of interest? Any realizable digital machine must have finite storage capabilities and finite word size for input data samples or computation results. Thus in practical applications it is always necessary to represent input and output data within a bounded set of numbers of finite representation. If the input is scaled and the resulting output is correspondingly rescaled, then all data computation within a digital machine can be represented as being performed on a finite set of integers. With suitable modulo arithmetic defined to satisfy closure, this set of integers forms a subset of a finite field or ring to which NTT concepts may be applied.

To have all required input and output quantities representable, they must belong to Z_M. Thus, with inclusion of signed numbers, it is necessary that the convolved sequences $h(n)$ and $x(n)$ be properly scaled so that

$$|y(n)| \leqslant \frac{M}{2} \ ,$$

where $y(n) = h(n) * x(n)$, in a manner similar to overflow constraints in fixed-point arithmetic. More generally, the range of $y(n)$ must be limited to an interval of less than M. Without this scaling, aliasing in amplitude, which is analogous to the aliasing in the frequency domain associated with discrete time representations, can occur. With proper scaling the results of convolution in the ring of integers modulo M are the same as these with ordinary arithmetic [18].

ARITHMETIC MODULO M

Prior to presenting a definition of the NTT we will need to discuss applicable arithmetic modulo M properties from number theory. This modular arithmetic also provides the basis for a discussion of conveniently applied conditions on the roots of unity α, convolution length N, and modulus M.

The number theory utilized with respect to the NTT is available in any basic book on the subject, such as Refs. 21 and 22. If

$$a = b + kM, \qquad 0 \leqslant b \leqslant M - 1,$$

then a is said to be congruent to b modulo M. It can be seen that b is the remainder when a is divided by M. This congruence relation is often also represented as $a \equiv b$, $a = b$ mod M, or $((a)) = b$. Any integer is congruent modulo M to exactly one integer in the ring Z_M, which was described previously. In general, modular arithmetic includes addition, subtraction, and multiplication. Division exists only if the inverse of the particular divisor exists. The results of all arithmetic must be expressed as a quantity within Z_M by means of a residue reduction. Within the above limitations modular arithmetic obeys the commutative, associative, and distributive laws and includes the identity, arithmetic inverse, closure, and analogy properties [19; 21, Ch. 9].

The basic approach to computation of circular convolution via the NTT is that as long as the final results can be properly represented within the quantities available in Z_M, the final congruence relation will represent an equality with the convolution results using ordinary arithmetic. Thus, in spite of overflows during intermediate steps of the computation, the results modulo M will be the same as those obtained by ordinary arithmetic. This same philosophy is used quite commonly with respect to 1's and 2's complement binary arithmetic.

Easily applied relationships among α, N, and M can be developed in order to properly utilize the number theoretic transform. If the requirement is to perform a particular convolution on a specified digital machine, these relationships can be used to select the most efficient NTT. Using number theory, we will consider such relationships here.

An important consideration in rings of integers modulo M is the existence of the inverse of a quantity p belonging to Z_M. An inverse of p exists in Z_M if and only if p and M are relatively prime [19; 22, p. 51]. The existence of an inverse transform will require that α have an inverse. Thus α and M must be relatively prime, expressed as $(\alpha, M) = 1$. Likewise, for N^{-1} to exist, $(N, M) = 1$.

For $(\alpha, M) = 1$, let $\varphi(M)$ be the number of integers in Z_M relatively prime to M. If the unique prime factors of M are such that $M = p_1^{r_1} p_2^{r_2} \ldots p_\ell^{r_\ell}$, then $\varphi(M) = M(1 - 1/p_1)(1 - 1/p_2) \ldots (1 - 1/p_\ell)$ [21, p. 111]. It can be shown that

$$\alpha^{\varphi(M)} = 1 \bmod M,$$

which is known as Euler's theorem [21, Ch. 12].

Analogous to the DFT, the powers of α mod M must form a sequence of period N with $\alpha^N = 1 \bmod M$. From Euler's theorem it can be seen that the powers of α will be periodic with period of at most $\varphi(M)$. For M a prime, $\varphi(M) = M - 1$ and

$$\alpha^{M-1} = 1 \bmod M,$$

which is known as Fermat's theorem. For certain values of α, known as primitive roots, the period will equal $\varphi(M)$. Thus the powers of a primitive root generate the total set of nonzero elements in Z_M [20]. For composite M, the period of the sequence of powers of α will be less than $M - 1$. The period is the smallest N for which $\alpha^N = 1 \bmod M$. Since $\varphi(M) \geqslant N$, $\varphi(M)$ can be represented as $\varphi(M) = aN + b$ where $0 \leqslant b < N$. If we apply Euler's theorem, $\alpha^{aN+b} = 1 \bmod M$, and since $\alpha^N = 1 \bmod M$, $\alpha^b = 1 \bmod M$. Since N is the smallest integer for which $\alpha^N = 1 \bmod M$, b must be 0 and therefore $N | \varphi(M)$.

If M has the unique prime power factorization $M = p_1^{r_1} p_2^{r_2} \ldots p_\ell^{r_\ell}$, with p_ℓ distinct primes, it can be shown [18] that a necessary and sufficient condition for a transform of the DFT form with the CCP property to exist is $N | 0(M)$ where $0(M) \triangleq gcd\ \{p_1 - 1, p_2 - 1, \ldots, p_\ell - 1\}$. Thus the maximum transform length is $N_{\max} = 0(M)$ [20].

NUMBER THEORETIC TRANSFORM

If the NTT is expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n)\alpha^{nk}, \qquad k = 0, 1, \ldots, N-1,$$

an inverse transform can be defined as

$$\lambda(n) = N^{-1} \sum_{k=0}^{N-1} X(k)\alpha^{-nk}, \qquad n = 0, 1, \ldots, N-1$$

where the exponents of α can be evaluated mod N, since $\alpha^N = 1 \bmod M$, and all other arithmetic is performed mod M. The inverse transform relation is of course valid only insofar as it can be shown to produce the original sequence $x(n)$ from the NTT sequence $X(k)$. This can be shown by substituting the transform sum into the inverse transform expression, as follows [16]:

$$\lambda(n) = N^{-1} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} x(m)\alpha^{mk} \right) \alpha^{-nk}, \qquad n = 0, 1, ..., N-1.$$

Interchanging the order of summation results in

$$\lambda(n) = \sum_{m=0}^{N-1} x(m) \left(N^{-1} \sum_{k=0}^{N-1} \alpha^{(m-n)k} \right).$$

Let $i = m - n$ and consider the cases $i = 0 \bmod N$ and $i \neq 0 \bmod N$. First it is obvious that

$$N^{-1} \sum_{k=0}^{N-1} \alpha^{ik} = 1, \qquad i = 0 \bmod N.$$

Second, since

$$(1 - \alpha^{i})N^{-1} \sum_{k=0}^{N-1} \alpha^{ik} = N^{-1}(1 - \alpha^{Ni}) = 0, \qquad i \neq 0 \bmod N$$

it follows that if $(1 - \alpha^{i})$ has a multiplicative inverse, then

$$N^{-1} \sum_{k=0}^{N-1} \alpha^{ik} = 0, \qquad i \neq 0 \bmod N$$

and $\lambda(n) = x(n)$. Thus the inverse NTT, presented above, is indeed valid.

From the above consideration of the relation between the NTT and its inverse, several restraints on α, N, and M are apparent. In satisfying the transform pair relationship, it was necessary that α be a root of unity of order N and the multiplicative inverses of α, N, and $1 - \alpha^{i}$, $i \neq 0 \bmod N$, exist. It was shown here earlier that, in a ring of integers modulo an integer M, a quantity p has a multiplicative inverse only if $(p, M) = 1$. Thus the restraints on α, N, and M can be listed as

$$\alpha^{N} = 1 \bmod M, \text{ with } N \text{ the smallest such integer,}$$

$$(N, M) = 1,$$

$$(1 - \alpha^{i}, M) = 1, \qquad i \neq 0 \bmod N.$$

It is important to note that the NTT itself provides no meaningful information and its usefulness is only with respect to the CCP property. Consider two sequences $a(n)$, $b(n)$ with NTTs $A(k)$, $B(k)$. If $C(k) = A(k)B(k)$, then

$$c(n) = N^{-1} \sum_{k=0}^{N-1} A(k)B(k)\alpha^{-nk}$$

or

$$c(n) = N^{-1} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} a(m)\alpha^{mk} \right) \left(\sum_{\ell=0}^{N-1} b(\ell)\alpha^{\ell k} \right) \alpha^{-nk}.$$

If the commutative, associative, and distributive laws are assumed to hold in the field or ring under consideration, then

$$c(n) = \sum_{m=0}^{N-1} \sum_{\ell=0}^{N-1} a(m)b(\ell) \left(N^{-1} \sum_{k=0}^{N-1} \alpha^{(m+\ell-n)k} \right).$$

Use of the results for

$$N^{-1} \sum_{k=0}^{N-1} \alpha^{i}$$

as defined earlier with respect to the NTT and its inverse results in

$$N^{-1} \sum_{k=0}^{N-1} \alpha^{(m+\ell-n)k} = 1, \qquad \ell = n - m \bmod N$$

which is otherwise 0. Therefore, with the same restraints on α, N, and M presented earlier,

$$c(n) = \sum_{m=0}^{N-1} a(m)b(n - m \bmod N).$$

Thus $c(n)$, the inverse NTT of the product of the NTTs of $a(n)$ and $b(n)$, is shown to be congruent to the circular convolution of $a(n)$ and $b(n)$. If $|a(n)|$, $|b(n)|$ are scaled so that $|c(n)| < M/2$, then the results of the NTT convolution modulo M will be equivalent to those of circular convolution computed directly with ordinary arithmetic.

MODULUS M

From the discussion up to this point, it can be seen that given a modulus M the possible lengths N for an NTT circular convolution are exactly those for which $N|0(M)$ with $N_{\max} = 0(M)$. In determining the particular NTT to apply in a given situation, the most convenient approach would be to choose M and then determine N and α. In considering these parameters of the NTT, we must consider the efficiency of the computation in relationship to other methods of achieving circular convolution. The present discussion is directed toward implementation of the NTT on a binary computing device.

Several factors must be considered in choosing M. Its value should provide a binary word length large enough to contain the results of the convolution. For example, if two sequences of length N with maximum amplitudes represented within b bits are convolved, then $\log_2 M > 2b + \log_2 N$. The form of M should be such that residue reduction is a simple operation not requiring division.

Having chosen M, the corresponding value of N should ideally be a power of 2 or at least highly composite so that the efficiency of an FFT algorithm can be applied to the computation. Finally, the value of α should provide for simple operations in implementing binary multiplications by powers of α. If α equals 2 or a power of 2, then multiplications by powers of α correspond to bit shifts. Other values of α such as -2 and $\sqrt{2}$ can also be used with some slight increase in computing complexity.

To perform arithmetic modulo M we must perform a residue reduction on the result of each computation. However, to also keep the required word size and overall storage requirements to a minimum, we must perform a residue reduction, as necessary, after all intermediate computations as well. Thus the efficiency with which residue reduction can be accomplished is an important consideration in choosing M. The simplest modulus for this purpose would be of the form 2^b, where b is any positive integer, and residue reduction only involves retaining the b least significant bits. However, since 2 is a prime factor, $N_{\max} = 1$ and 2^b is thus useless as a modulus. It is thus apparent that in order to attain any practical sequence lengths, M must be odd.

Next, a modulus of the form $2^b - 1$ is considered. The most interesting sequence lengths, in relation to word length, occur for b a prime. Numbers of the form $M_p = 2^p - 1$, p a prime, are known as Mersenne numbers and are of interest, since $2^p = 1 \bmod M_p$ and thus $\alpha = 2$ is a root of unity. Number theoretic transforms with modulus M_p (designated Mersenne number transforms (MNTs)) were first proposed by Rader [17]. Rader also showed that MNTs of $N = 2p$ with $\alpha = -2$ also exist. Arithmetic modulo M_p is quite simple. Residue reduction involves only adding the word formed by the bits beyond the p least significant bits to the word formed by the p least significant bits, since $2^p \equiv 1$, $2 \cdot 2^p \equiv 2$, etc. The limitation of the MNT, however, is that the values of N are not very composite and thus do not lend themselves to FFT computation procedures.

Consider a modulus of the form $2^b + 1$. For b odd, $3 | 2^b + 1$, and $N_{\max} = 2$. Therefore b must be even. It is found that an NTT with a modulus of the form $F_t = 2^b + 1$, $b = 2^t$, known as a Fermat number, is of the most interest and is known as a Fermat number transform (FNT). Transforms in a ring of integers modulo a Fermat number were first suggested by Rader [17]. These transforms were considered by Agarwal and Burrus [18,20], who defined them mathematically and discussed their implementation and application to discrete convolution. The F_t are prime for $t = 0, 1, ..., 4$ and factorable with a prime factor of the form $K2^{t+2} + 1$ for $t > 4$ [18]. Thus for $t \leqslant 4$, $0(F_t) = 2^b$, $b = 2^t$, and valid NTT exist for all $N = 2^c$, $c \leqslant b$. For $t > 4$, $2^{t+2} | 0(F_t)$ and valid NTT result for all $N = 2^c$, $c \leqslant t + 2$. For $t = 5,6$ which correspond to many practical signal processing applications, $N_{\max} = 0(F_t) = 2^{t+2} = 4b$ [18]. Thus, for the FNT, the resulting N are practical values that are integral powers of 2. Therefore, an FFT type of computational procedure exists. This procedure can be any one of the available FFT algorithms [9] with powers of $\exp(-j2\pi/N)$ replaced by powers of α.

Since $F_t = 2^b + 1$, it follows that $2^b = -1 \bmod F_t$. Then $2^{2b} = 1 \bmod F_t$ and $(\sqrt{2})^{4b} = 1 \bmod F_t$ [18]. Therefore, $N = 2b$ requires that $\alpha = 2$, and $N = 4b$ corresponds to $\alpha = \sqrt{2}$. In the FNT, with an FFT realization, computation with $\alpha = \sqrt{2}$ requires only a small increase in complexity, since odd powers of α would occur in either the first or the last stage of a particular FFT algorithm. It can be seen that the word length required for the FNT is $b + 1$. The most significant bit is required only to represent 2^b. Residue reduction is achieved in a manner similar to that described earlier for the MNT. However, since $2^b = -1 \bmod M$, the word formed from the quantity beyond the b least significant bits is subtracted from the word formed by the b least significant bits.

In the present discussion the primary interest is in the FNT with an α of 2, or an integer power of 2, which is known as the Rader transform (RT). The RT, with $\alpha = 2$ and $N = 2^m$, is defined as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n)2^{nk} \bmod F_t, \qquad k = 0, 1, ..., N-1,$$

$$x(n) = 2^{-m} \sum_{k=0}^{N-1} X(k)2^{-nk} \bmod F_t, \qquad n = 0, 1, ..., N-1.$$

The RT appears to be the optimum NTT that results from all those available among the various M. In addition to the general advantages of the NTT due to N that are integral powers of 2, in the RT all multiplications by powers of α involve powers of 2. Thus these multiplications are each performed by a simple binary word shift followed by a subtraction if a residue reduction is required.

IMPLEMENTATION OF FNT

The FNT has been described here and was shown to involve a modulus of the form $F_t = 2^b + 1$, $b = 2^t$. Thus the word length required for computation must be such that all integers from 0 to 2^b can be represented. This necessitates a word length of $b + 1$ bits. The most significant bit will be used only to represent 2^b when it occurs either in the data or in a computation result. Therefore, implementation of the FNT with b-bit arithmetic will result in some small input quantization error because of the need to approximate a data input sample of -1 by the closest available values, 0 or -2. If the value 2^b occurs as the result of a computation, all output data for the particular section or data block involved will be meaningless. For most applications the probability of such an occurrence is small enough to be acceptable. Agarwal and Burrus [18] investigated the implementation of the various arithmetic modulo F_t operations required for an FNT with a word length of b bits, describing the required operations as follows:

1. Negation
   Complement each bit and add 2 to the result.

2. Addition
   Add the two b-bit integers and end-around subtract any carry bit.

3. Subtraction
   Negate the subtrahend and add the result to the minuend.

4. General Multiplication

   Multiply and get $C_L + C_H 2^b$, where C_L and C_H are words formed from the b least and most significant bits, respectively. From C_L subtract C_H modulo F_t.

5. Multiplication by a power of 2

   Use double register of length $2b$

   To multiply by 2^k:

   Load data in lower half of $2b$-bit double register.
   Shift left k positions and from C_L subtract C_H modulo F_t.

   To multiply by 2^{-k}:

   Load data in upper half of $2b$-bit double register.
   Shift right k positions and from C_H subtract C_L modulo F_t.

It should be noted that in the RT, the powers of α need not be stored or even computed. The exponents, which can be determined from the position in the procedure, are the bit shifts needed to accomplish the multiplications required in the transform.

TRANSFORM LENGTH LIMITATIONS

The transform lengths for the NTT, and thus the lengths of circular convolutions that can be achieved, are proportional to the word length b. Thus to consider long transform lengths requires that a large word size be used in performing computations. Table 1 indicates, for various practical values of b, the available values of N corresponding to $\alpha = 2$ (RT) and $\alpha = \sqrt{2}$ as well as the $N_{\max}$ and related α. It can be seen that in order to achieve a transform of length $N = 128$, using the RT, a word length of 64 bits is required. For applications using a minicomputer, such a word length would be totally restrictive. It would, however, be possible to work with such transform lengths, with some storage and speed inefficiencies, on a 32-bit machine with double precision such as the IBM 360 or 370 series. In the case of a special purpose machine where it was most important to accomplish such transform lengths, the long word lengths might be acceptable.

Table 1 — Circular Convolution Lengths for Various Word Lengths*

Word Length, b	F_t	Convolution Length N		$N_{\max}$	α for $N_{\max}$
		($\alpha = 2$)	($\alpha = \sqrt{2}$)		
8	$2^8 + 1$	16	32	256	3
16	$2^{16} + 1$	32	64	65536	3
32	$2^{32} + 1$	64	128	128	$\sqrt{2}$
64	$2^{64} + 1$	128	256	256	$\sqrt{2}$

*From Ref. 18.

The word length restrictions of NTT were first considered by Rader [17] with respect to the MNT. Rader suggested that performing long one-dimensional convolutions by means of two-dimensional NTT could relieve these restraints. The computation of a one-dimensional discrete convolution by means of two-dimensional FNT techniques was analyzed by Agarwal and Burrus [23]. Their analysis reveals that a discrete one-dimensional circular convolution of length $N = L \cdot M$ can be achieved by two-dimensional ($2L \times M$) transforms. These procedures involve an expansion from a one- to a two-dimensional convolution by repetition of sequence samples and inclusion of additional zero samples in a manner analogous to performing linear convolution by circular convolution. The word length requirement with two-dimensional transform techniques is proportional to the square root of sequence length ($\sqrt{N}$) as compared to the one-dimensional case with a proportionality to N.

The improvement in the magnitude of the lengths of convolutions that can be handled with the FNT is dramatic, as can be seen from a comparison of the N available with two-dimensional transforms, indicated in Table 2, and the one-dimensional case of Table 1. The value of $N_{\max}$ is $8b^2$ as compared to $4b$. In performing two-dimensional convolution, the FNT can be used along the long dimension with a compatible N and a direct or other efficient convolution procedure used on the short dimension.

Table 2 — One-Dimensional Circular Convolution Lengths Available with Two-Dimensional FNT or RT*

Word Length, b	$N(\alpha = 2)$	$N(\alpha = \sqrt{2})$
16	512	2048
32	2048	8192
64	8192	32768

*From Ref. 18.

Related to the word length restraint is the problem of providing a modulus large enough to contain the magnitude of the resulting convolution. This can be quite a limiting factor with respect to implementing the FNT on a minicomputer. One technique for working with shorter word lengths involves dividing the words approximately in half into shorter segments and convolving with half the number of bits, but it requires three convolutions as compared to one for full word length [18]. Another technique involves working with two different smaller moduli, M_1, M_2 such that $M_1 \cdot M_2 = M$, and combining the final result mod $M_1 M_2$ by the Chinese remainder theorem [20].

COMPLEX NUMBER THEORETIC TRANSFORMS

Up to this point the discussion has been limited to the application of the NTT to the convolution of real number sequences. In many practical applications, such as in radar and communications, the sequences to be convolved can consist of complex quantities. If each of these complex numbers is treated in rectangular form as consisting of a real and an imaginary part these can be scaled, as in the real NTT, such that they are represented in a digital computing device as members of a set of integers with finite bounds. In the complex case, then, the numbers that result consist of integer real and imaginary parts and are known as complex or Gaussian integers, which are described in Ref. 22 (pp. 178, 179).

The concepts and advantages in computation efficiency and the lack of round-off error discussed with respect to the real number NTT are applicable to the complex number theoretic transform (CNT). The extension of number theoretic techniques to the convolution of complex sequences has only been considered quite recently. Reed and Truong [24] discuss the extension of the NTT to complex sequences using transforms defined over a Galois field $GF(M^2)$ when M is prime such that -1 is a quadratic nonresidue, i.e. when $x^2 = -1 \bmod M$ has no solution in $GF(M)$. Agarwal and Burrus [20] discuss the CNT in a complex integer field Z_M^c, which is $GF(M^2)$, and develop relations among the transform parameters analogous to those developed for real NTT. A more general approach to the CNT was taken by Vegh and Leibowitz [25] who define a complex NTT in a finite ring that simulates the complex integers.

The complex NTT has the same form as that presented earlier for the real-number case. In the complex case, however, the data sequence and the powers of α are complex integers of the form $a + jb$. These complex integers are members of a set Z_M^c where a, b are elements of Z_M. In order to prevent aliasing, the real and imaginary parts of the output of any computations modulo M must be individually bounded in magnitude by $M/2$, as in the case of the real NTT. If $x = a_1 + jb_1$ and if $y = a_2 + jb_2$, the operations of addition and multiplication modulo M required to carry out the computation of the CNT within a field or ring of integers modulo M are defined as follows:

$$x \oplus y = ((a_1 + a_2)) + j((b_1 + b_2))$$

$$x \odot y = ((a_1 a_2 - b_1 b_2)) + j((a_1 b_2 + a_2 b_1)).$$

More generally modulo M arithmetic involving $Z = a + jb$, where a, b are any integers for $[[Z]] = ((\mathrm{Re}\, Z)) + j((\mathrm{Im}\, Z))$ is defined as

$$[[Z_1 + Z_2]] = [[Z_1]] \oplus [[Z_2]]$$

$$[[Z_1 Z_2]] = [[Z_1]] \odot [[Z_2]].$$

The discussion of Reed and Truong [24] as well as that of Agarwal and Burrus [20] considers the case where M is prime and Z_M^c and Z_M correspond to the finite fields $GF(M^2)$ and $GF(M)$, respectively. The existence of the field Z_M^c (i.e., - 1 is a quadratic nonresidue modulo M) requires that a root of 1 of order 4 not exist in Z_M, or $4 \nmid 0(M) = M - 1$. Since $4 \nmid M_p - 1 = 2^p - 2$, CNT exist for Mersenne numbers in the case of fields. For Fermat numbers, $4 | F_t - 1 = 2^{2^t}$ and no corresponding CNT exist. The complex NTT with the CCP property in Z_M^c will exist if and only if $N | M^2 - 1$ [20]. In both Refs. 20 and 24, procedures for finding primitive elements for α are presented, and Ref. 24 provides a table of the simplest prime elements whose use should lead to the least hardware. The CNT defined in the finite field $GF(M^2)$ lead to transforms with the CCP property, but multiplication by powers of α is not as simple as that of the RT discussed earlier.

Since $N|M^2 - 1$, maximum period $N_{\max}$ of α of $GF(M^2)$ for M a Mersenne prime is

$$N_{\max} = M^2 - 1 = (2^p - 1)^2 - 1 = 2^{p+1}(2^{p-1} - 1).$$

Therefore, any $N = 2^k$ for $1 \leqslant k \leqslant p + 1$ divides $M^2 - 1$ and can be the transform length for a CNT in a field with a power-of-2 FFT algorithm applicable to the computation.

The overflow problem for convolutions with the CNT is considered by Reed and Truong [24]. The real and imaginary parts of the data sequences must be limited in magnitude to

$$A = \left[\sqrt{\frac{M-1}{4N}}\right],$$

where $[x]$ denotes the greatest integer less than x, in order for the CNT to lead to the correct result for circular convolution.

The more general case of convolution with a CNT defined in the ring of complex integers with the real and imaginary parts modulo an integer M is considered by Vegh and Leibowitz [25]. With the restraints for limiting the magnitude of the real and imaginary components and with complex modulo arithmetic as described earlier, CNT in such rings are defined that possess the CCP property. As in the real NTT, in order to satisfy the requirements for a CNT and have a suitable inverse transform the inverses of N and $1 - 2^t$, $t \neq 0 \bmod N$ must exist in the ring and α must be of order N. A complex Mersenne transform can then be defined with $M = M_p = 2^p - 1$, p prime, $N = p$ and $N^{-1} = (1 - M_p)/p$. To give an example of the Mersenne CNT, with input sequence $z(n) = a(n) + jb(n)$, we define a transform as

$$Z(k) = \left[\left[\sum_{n=0}^{N-1} z(n)2^{nk}\right]\right] = \left(\left(\sum_{n=0}^{N-1} a(n)2^{nk}\right)\right) + j\left(\left(\sum_{n=0}^{N-1} b(n)2^{nk}\right)\right),$$

with inverse transform

$$z(n) = \left[\left[N^{-1}\sum_{k=0}^{N-1} Z(k)2^{-nk}\right]\right] = \left(\left(N^{-1}\sum_{k=0}^{N-1} A(k)2^{-nk}\right)\right)$$

$$+ j\left(\left(N^{-1}\sum_{k=0}^{N-1} B(k)2^{-nk}\right)\right),$$

where the transform domain sequence is $Z(k) = A(k) + jB(k)$. The real and imaginary parts of the transform are NTTs of the real and imaginary parts of the input sequence, respectively. Likewise, the real and imaginary parts of the inverse transform are inverse NTTs of the real and imaginary parts of the transform sequence, respectively. Thus these CNT can be carried out with binary word shifts and additions but no multiplications. With $\alpha = 2$ the transform length is limited to p. For the MNT, Rader [17] showed that, with $\alpha = -2$, NTTs with an N of $2p$ exist. In the complex Mersenne transform, sequence lengths of $4p$ and $8p$ can be attained with complex α of $2j$ and $1 + j$, respectively. The computations involved with the use of such α are not difficult. For example the powers of α, represented as α^t, are of the form $2^s a + j2^s b$ where a, b are 0 or ± 1 and $s = t$ for $\alpha = 2j$, while $s = [t/2]$ or the integer part of $t/2$ for $\alpha = 1 + j$. Complex number theoretic transforms defined in a ring of integers modulo M result in multiplications by powers of α that are simpler than those associated with the CNT in a field. A complex Fermat number transform can also be shown to exist with the additional advantage that N is a power of 2 and thus the efficiency of an FFT-type computation procedure can be utilized.

It should be noted that the sequence length limitations described earlier for the NTT apply to the CNT. As in the NTT case, these concepts are applicable to longer sequences by the multidimensional procedures described in Ref. 23.

The disclosure by Vegh and Leibowitz [25] came as a result of the present author's suggestion that the DFT might be efficiently computed using NTT convolution. This could either provide a faster means of computing the DFT or at least permit a device that computed an NTT to compute a DFT without arithmetic round-off error. The basis for the computation of the DFT by NTT methods lies in the chirp z-transform (CZT) algorithm of Bluestein [26]. For a sequence $x(n)$ of length N, the DFT can be represented as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N-1} x(n)(W_N^{1/2})^{2nk} = (W_N^{1/2})^{k^2} \sum_{n=0}^{N-1} x(n)(W_N^{1/2})^{n^2}(W_N^{-1/2})^{(k-n)^2}$$

where $W_N = \exp(-j2\pi/N)$. Thus the DFT can be computed by weighting $x(n)$ by $W_N^{n^2/2}$ in the time domain, convolving the resulting sequence with $W_N^{-n^2/2}$, and weighting the resulting convolution by $W_N^{k^2/2}$ in the frequency domain. The powers of W_N are in general complex. The convolution required for this computation can thus be computed by CNT techniques. All computations can be performed with modulo arithmetic as long as the magnitudes of real and imaginary parts of the final computation are within the bound of $M/2$. An example of such a DFT computation is presented in Ref. [25].

NTT vs FFT CONVOLUTION

At the present time applications of discrete convolution and correlation are performed most commonly by efficient FFT techniques. It is therefore of considerable interest to compare the efficiency, including the number and type of operations, of NTT and FFT convolution. The most efficient form of the NTT is the RT ($\alpha = 2$). Comparisons between the RT and FFT were considered by Agarwal and Burrus [18].

Because of the word length required to represent a convolution without roundoff, the RT requires approximately twice the number of bits as the FFT. Since the FFT requires a real and an imaginary part, two words are required for each sequence value. Therefore, the number of bits per data point and thus the hardware requirements are about the same for the two transforms. For the FFT, with real data, symmetry properties permit two transforms to be performed with one complex transform [7]. This requires some additional computation but can reduce the overall computation time. In the case of convolution with complex data, the CNT involves two computations of the RT. This requires twice the computation time of a single RT. With twice the hardware, these

RT can be performed simultaneously, and thus the CNT can be performed in the time of a single RT.

Agarwal and Burrus [18] assume a straightforward FFT with $b/2$-bit real and imaginary parts and a RT with b-bit modulo arithmetic. A $b/2$-bit complex addition/subtraction is equivalent to a b-bit addition/subtraction modulo F_t. A $b/2$-bit complex multiplication is of approximately the same complexity as that of a b-bit multiplication modulo F_t. The dramatic advantage with the RT is with respect to multiplication by powers of α, which are implemented as bit shifts and subtractions, which is far simpler than the complex multiplications required in the FFT.

For an N-point FFT in the complex number field, the number of operations consists of N complex additions/subtractions in each iteration, or a total of $N \log_2 N$ plus $N/2$ complex multiplications in each array after the first, or a total of $(N/2) \log_2 (N/2)$. As described earlier, with N an integer power of 2, an NTT with a Fermat number modulus can be performed using any FFT algorithm with W_N replaced by α. Thus, the number of modular additions/subtractions and multiplications will be the same as that in the complex number field. The multiplications in the case with $\alpha = 2$ involve powers of 2 which can be performed efficiently by bit shifts and subtractions. In either implementation, the convolution of two sequences involves the inverse transform of the product of their transforms. The complexity of the N-transform-domain products and of the multiplications by N^{-1} required in the inverse transforms is approximately the same in both cases.

Several other points of comparison can be considered. The FFT requires storing the powers of W_N, which is unnecessary in the RT. Since the RT involves no round-off error, there will be no round-off noise in the output of digital signal processing systems implemented with this technique.

Agarwal and Burrus [18] implemented both FFT and FNT convolutions on an IBM 370/155. The FFT used involved a highly efficient mixed radix algorithm. The FNT procedures included the RT ($\alpha = 2$) plus $\alpha = \sqrt{2}$, as well as a two-dimensional RT. The results of timings for the various implementations indicate speed improvements on the order of 3 to 5 for the FNT over the FFT. It should be noted that the computer used for the comparison does not do arithmetic modulo Fermat numbers and has unusually high-speed multiplication relative to addition. It would appear then that in practical applications with special purpose hardware including a modular arithmetic unit or where multiplication is generally much slower than addition, much greater speed advantages would result from number theoretic methods.

ACKNOWLEDGMENTS

The support of the author's research efforts by Mr. R.E. Ellis, Head of the Special Projects Organization, NRL, and the guidance and advice of Prof. N. Kyriakopoulos of George Washington University are greatly appreciated. The discussions of this material with Dr. E. Vegh of the Advanced Concepts Staff, NRL, have been invaluable to the author's understanding of the number theory employed in this field of study. The author wishes to thank Prof. C.S. Burrus of Rice University for the opportunity to discuss the NTT with him on several occasions. The work of Mrs. A. Latham with respect to preparation of the manuscript is greatly appreciated.

REFERENCES

1. L.M. Leibowitz, "Overview of Digital Signal Processing Theory," NRL Report 7870, May 20, 1975.
2. T.G. Stockham, Jr., "High-Speed Convolution and Correlation," *1966 Spring Point Computer Conf., AFIPS Proc.* 28, 229-233. Spartan, Washington, D.C., 1966.
3. H.D. Helms, "Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters," *IEEE Trans.* AU-15, 85-90 (June 1967).
4. C.S. Burrus, "Block Implementation of Digital Filters," *IEEE Trans.* CT-18, 697-701 (Nov. 1971).
5. C.S. Burrus, "Block Realization of Digital Filters," *IEEE Trans.* AU-20, 230-235 (Oct. 1972).
6. J.W. Cooley, P.A.W. Lewis, and P.D. Welch, "Application of the Fast Fourier Transform to Computation of Fourier Integrals, Fourier Series, and Convolution Integrals," *IEEE Trans.* AU-15, 79-84 (June 1967).
7. J.W. Cooley, P.A.W. Lewis, and P.D. Welch, "The Finite Fourier Transform," *IEEE Trans.* AU-17, 77-85 (June 1969).
8. J.W. Cooley, and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Compu.* 19, 297-301 (Apr. 1965).
9. E.O. Brigham, *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
10. C.M. Rader, "An Improved Algorithm for High Speed Autocorrelation with Applications to Spectral Estimation," *IEEE Trans.* AU-18, 439-441 (Dec. 1970).
11. P.D. Welch, "A Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Trans.* AU-17, 151-157 (June 1969).
12. W.M. Gentleman and G. Sande, "Fast Fourier Transform—for Fun and Profit," *1966 Spring Joint Computer Conf., AFIPS Proc.* 29, 563-558. Spartan, Washington, D.C., 1966.
13. C.J. Weinstein, "Roundoff Noise in Floating Point Fast Fourier Transform Computation," *IEEE Trans.* AU-17, 209-215 (Sept. 1969).
14. T. Kaneko and B. Liu, "Accumulation of Round-off Error in Fast Fourier Transforms," *J. Ass. Comput. Mach.* 17, 637-654 (Oct. 1970).
15. D.A. Pitassi, "Fast Convolution Using the Walsh Transform," *Proc. Symposium,* Applications of Walsh Functions, 2d ed., Washington, D.C., Apr. 1971, pp. 130-133.
16. J.M. Pollard, "The Fast Fourier Transform in a Finite Field," *Math. Comput.* 25, 365-374 (Apr. 1971).
17. C.M. Rader, "Discrete Convolutions via Mersenne Transforms," *IEEE Trans.* C-21, 1269-1273 (Dec. 1972).
18. R.C. Agarwal and C.S. Burrus, "Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering," *IEEE Trans.* ASSP-22, 87-97 (Apr. 1974).

19. C.M. Rader, "Convolution and Correlation Using Number Theoretic Transforms," Section 6.19 *of Theory and Application of Digital Signal Processing,* by L.R. Rabiner and B. Gold, Prentice-Hall, Englewood Cliffs, N.J., 1975.

20. R.C. Agarwal and C.S. Burrus, "Number Theoretic Transforms to Implement Fast Digital Convolution," *Proc. IEEE,* 63, 550-560 (April 1975).

21. O. Ore, "Number Theory and Its History," McGraw-Hill, New York, 1948.

22. G.H. Hardy and E.M. Wright, "An Introduction to The Theory of Numbers," 4th ed., Oxford Univ. Press, Oxford, England, 1960.

23. R.C. Agarwal and C.S. Burrus, "Fast One-Dimensional Digital Convolution by Multidimensional Techniques," *IEEE Trans.* ASSP-22, 1-10 (Feb. 1974).

24. I.S. Reed and T.K. Truong, "The Use of Finite Fields to Compute Convolutions," *IEEE Trans.* IT-21, 208-213 (Mar. 1975).

25. E. Vegh and L.M. Leibowitz, "Fast Complex Convolution Using Number Theoretic Transforms," submitted to the IEEE Transactions on Acoustics, Speech and Signal Processing, April 1975.

26. L.I. Bluestein, "A Linear Filtering Approach to the Computation of Discrete Fourier Transform," *IEEE Trans.* AU-18, 451-455 (Dec. 1970).